Addressing Accessibility: 3rd generation, ÆGIS, and Sun's approach

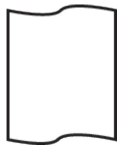
Peter Korn
Accessibility Architect
& Principal Engineer
Sun Microsystems, Inc.

20 November 2009

Overview

- Sun's accessibility approach & vision
  - Who am I, how did I get into the field?
- The ÆGIS project → accessibility work funded by the European Commission
- Thoughts about the National Broadband Plan & serving Americans with Disabilities

Urgency



- Every second...
  - 4 babies are born
  - 36 cell phones are activated
  - 411 web pages are created
- 1.3+ bn cell phones were sold in 2008
- 12.8+ bn new web pages appeared in 2008
- Google: indexes over 1 trillion web pages -

  http://news.cnet.com/8301-1023_3-9999814-93.html?tag=nefd.top

How many are accessible? How many in 2009? 2012?

1st theme: Built-in vs. Bolt-on

- Working with mature platforms means accessibility comes late to the table, is never really part of the underlying design

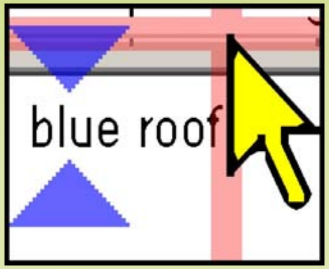
- Working with young platforms means making a gamble - will the platform be important enough in the future?

2nd theme: Evolution of access

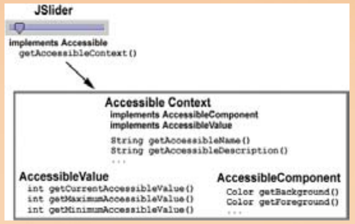
From simple terminal applications to accessible network desktops

1st Generation: late 1960s- early 1980s

2nd Generation: late 1980s – early 2000s



3rd Generation: 1997 and onward



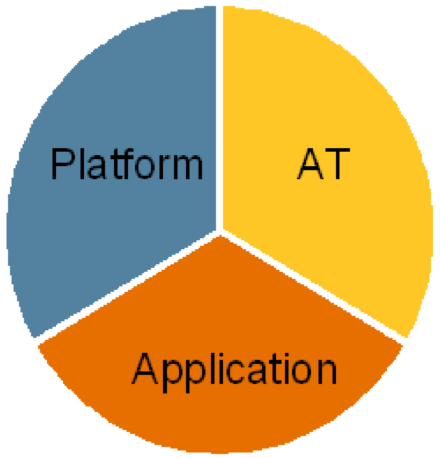
```
implements Accessible
  getAccessibleContext()

Accessible Context
  implements AccessibleComponent
  implements AccessibleValue
  String getAccessibleName()
  String getAccessibleDescription()
  ...

AccessibleValue
  int getCurrentAccessibleValue()
  int getMaximumAccessibleValue()
  int getMinimumAccessibleValue()

AccessibleComponent
  Color getBackground()
  Color getForeground()
  ...
```

3rd Generation enables thin client

Advances in capabilities, sophistication

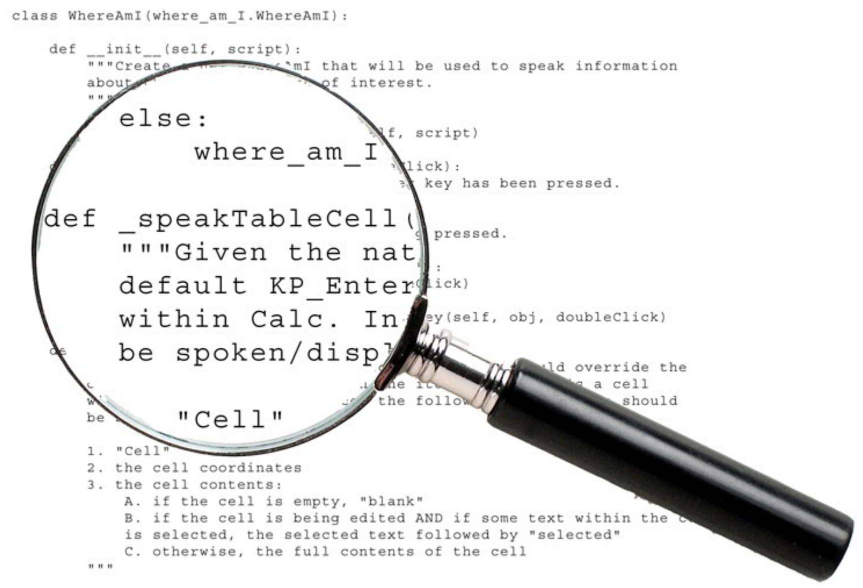
3rd theme: Divide the tasks

- First & second gen., AT had to do everything
  - Get the text, determine context (from buffer or OSM), magnify text
  - Special-case applications (MS-Office, Internet Explorer)
  - Create specialized drivers for specialized hardware
- The climate has changed
  - Greater awareness of people with disabilities
  - Laws worldwide requiring accessibility
- Proposal: divide the work into three pieces:
  - Platform: define, implement accessibility architecture
  - Application: support the platform accessibility arch.
  - AT: focus on the user experience



4th theme: Open source access

- All source code available for examination
- AT developers can fix their own bugs, release their own patches
- AT may be open source too
- Vendors and users can control their own destiny



```
class WhereAmI(where_am_I.WhereAmI):

    def __init__(self, script):
        """Create a [illegible]mI that will be used to speak information
        about [illegible] of interest.
        """
            [illegible]f, script)

            [illegible]lick):
            [illegible] key has been pressed.

            [illegible] pressed.

            [illegible]:
            [illegible]ick)
            [illegible]ey(self, obj, doubleClick)

            [illegible]ld override the
            [illegible] a cell
            [illegible] the follow[illegible] should
        be [illegible]

        1. "Cell"
        2. the cell coordinates
        3. the cell contents:
            A. if the cell is empty, "blank"
            B. if the cell is being edited AND if some text within the c
            is selected, the selected text followed by "selected"
            C. otherwise, the full contents of the cell
        """
```

```
else:
    where_am_I

def _speakTableCell(
    """Given the nat
    default KP_Enter
    within Calc. In
    be spoken/displ

       "Cell"
```

Who am I, how did I get here?

- **Started with the company that invented 2nd generation accessibility → Berkeley Systems**
- **Berkeley Systems started with 2 National Eye Institute Grants**
  - inLARGE → first graphical screen magnifier, 1987
  - outSPOKEN → first graphical screen reader, 1989
  - Sold for $195 & $495 respectively
- **Joined in 1992 to work on “cross-platform” version of outSPOKEN, for Mac & Windows (and SunOS)**
- **Pushed Apple, Microsoft for accessibility APIs**
  - Moved to Sun to build them into Java

Process of developing an accessible world

Physical world

CREATION

1. Define what "accessible" means
- a. how wide must a door be for a wheelchair to fit through it?
- b. how much force must you need to open a window?
- c. how do we make an elevator accessible - tones, Braille...

2. Create stock building materials that meet "accessible definition"
- a. build a set of standard doors - all wide enough for a wheelchair
- b. build a set of standard windows - with little force needed to open
- c. build a set of standard elevators - with tones, Braille, tactile symbols

3. Create tools for building accessible homes; using stock materials
- a. manuals & standards for how to install windows, doors, elevators
- b. specs. for wheelchair ramp construction; testing ramp elevation
- c. special tools for installing windows, doors, elevators, etc.

USE

4. Locate the building where it will work
- a. is the building near public transit?
- b. is there a wheelchair ramp leading up to the building?
- c. can people find the crosswalk buttons

5. Make accessible buildings, spaces

6. Disseminate the devices people need to use the accessible world
- a. distribute wheelchairs (that work with the ramps)
- b. provide canes for the blind, train seeing eye dogs
- c. diagnose hearing problems, prescribe hearing aids

Computer world

CREATION

1. Define what "accessible" means
- a. define keyboard navigation scheme
- b. define theme mechanisms for high contrast, large print
- c. define an accessibility API for communication with AT

2. Create stock UI component toolkits that implement "accessible"
- a. build sets of desktop UI components - menus, windows, etc.
- b. build sets of web UI components - charts, drag & drop, etc.
- c. build sets of mobile UI components - text fields, radio buttons, etc.

3. Create developer tools for building apps with stock UI toolkits
- a. manuals & standards for how to make accessible applications
- b. developer tools that provide the stock accessible UI toolkits
- c. developer tools that flag inaccessible application designs

USE

4. Make the platforms accessible, able to run AT
- a. does the platform expose accessibility APIs from applications?
- b. can the user select a high contrast, large print theme?
- c. does the platform have text-to-speech, Braille, for ATs to use?

5. Make the software applications accessible

6. Disseminate assistive technologies to people who need them
- a. include a screen reader with the platform, or a free download
- b. include on on-screen keyboard w/platform, or a free download
- c. include AAC software with the platform, or a free download

Process of developing an accessible world

ÆGIS

CREATION

1. Define what “accessible” means
Desktop user requirements (also IAccessible2, AT-SPI, etc.)
Web application user requirements (also WAI ARIA, JA-API)
Mobile user requirements
Mobile environment API definition

2. Create stock UI component toolkits that implement “accessible”
Rich Internet application UI component sets
Mobile application UI component sets

3. Create developer tools for building apps with stock UI toolkits
Tools for making accessible applications with RIA toolkits
Tools for making accessible mobile applications

USE

4. Make the platforms accessible, able to run AT
Accessible web platform (web browser)
Accessible mobile platform (cell phone, PDA)

5. Make the software applications accessible
Accessible rich Internet web applications
Accessible mobile applications

6. Disseminate assistive technologies to people who need them
Real-time text for the deaf; on desktop (and mobile)
Support for inexpensive web cams for head/eye tracking
Five different assistive technologies for mobile devices

Computer world

CREATION

1. Define what “accessible” means
a. define keyboard navigation scheme
b. define theme mechanisms for high contrast, large print
c. define an accessibility API for communication with AT

2. Create stock UI component toolkits that implement “accessible”
a. build sets of desktop UI components - menus, windows, etc.
b. build sets of web UI components - charts, drag & drop, etc.
c. build sets of mobile UI components - text fields, radio buttons, etc.

3. Create developer tools for building apps with stock UI toolkits
a. manuals & standards for how to make accessible applications
b. developer tools that provide the stock accessible UI toolkits
c. developer tools that flag inaccessible application designs

USE

4. Make the platforms accessible, able to run AT
a. does the platform expose accessibility APIs from applications?
b. can the user select a high contrast, large print theme?
c. does the platform have text-to-speech, Braille, for ATs to use?

5. Make the software applications accessible

6. Disseminate assistive technologies to people who need them
a. include a screen reader with the platform, or a free download
b. include on on-screen keyboard w/platform, or a free download
c. include AAC software with the platform, or a free download

EC accessibility grants:

- 3.5 year Sun-led ***ÆGIS*** for €8.22M; €1.36M to Sun:
  - Outcome a: “New approaches and solutions for **deeply embedding generalized accessibility support** within future mainstream ICT-based products and services...”
  - http://www.aegis-project.eu
- 3 year ***ACCESSIBLE*** for €2.6M; €386k to Sun:
  - Outcome b: “New methods & tools for **computer simulation of the user interaction and computer-based validation frameworks** supporting developers for a11y...”
  - http://www.accessible-eu.org/

ÆGIS Consortium

- **Large Industrial partners:**
  - Sun, AOL, RIM, Foundation Vodafone Spain
- **Disability organizations:**
  - ACE Centre, SU-DART, European Platform for Rehabilitation, Royal National Institute for the Blind, ONCE Foundation
- **Research organizations (generally SMEs):**
  - CERTH/HIT, CERTH/ITI, Fraunhofer IAO, SingularLogic SA, Conncept Swiss, P50, Bluepoint Solutions
- **Universities:**
  - Czech Technical University, University of Cambridge, Catholic University of Leuven, University Polytechnic Madrid, University of Toronto Adaptive Technology Resource Centre

What does *ÆGIS* do?

- Builds accessibility into ***all stages*** of future mainstream ICT: the 3 steps of “creation” and of “use”
- Works in 3 distinct ICT areas, of differing levels of accessibility “maturity” (desktop, web, mobile)
  - Doing different work based on that maturity level
- Develops an “Open Accessibility Framework” that can be applied to ICT accessibility generally
  - And proving that OAF for desktop, web, mobile
- Addresses broad range of disability scenarios, needs
- Does the bulk of the work in open source

What does *ACCESSIBLE* do?

- Builds a suite of per-disability & per-guideline set accessibility assessment tools
- Builds disability simulation tools
- Integrates these into developer tools
  - > In particular, into the NetBeans IDE

Open Desktop accessibility

- Basics
  - Keyboard navigation, themeing, AccessX
- For significant vision impairment
  - Orca with speech, magnification, Braille
- For significant physical impairment
  - GOK, Dasher
- Availability
  - Ubuntu, Fedora, OpenSUSE, OpenSolaris

Magnification built into GNOME Shell

- Objective: build magnification into the future GNOME desktop
  - > Support existing magnification needs (e.g. Orca)
  - > Be a platform for “3rd generation” magnification, and support for future cognitive impairment AT
- Approach:
  - > Leverage commodity video hardware developed for gaming, exposed via standard UNIX APIs (COMPOSITE)
  - > Build into GNOME Shell, likely future GNOME desktop
  - > Meet most pressing need first (Orca magnification)
- Status: early demo – fast, smooth & part of GNOME Shell

OpenGazer as a video switch

- Long term goal: eye tracking with commodity web cams
  - > Ideally including typical cameras built into laptops, mobile phones, etc. (OLPC too!)
- Interim steps:
  - > Detect various forms of facial movement and use as a “video switch” (e.g. looking up, smiling, blinking)
  - > Detect head movement and use as a “head mouse”
- Status:
  - > Switch detection working well now, more refinement needed
  - > 1 dimension head tracking working

Odt2daisy: accessible talking books

- Shipping 2.0 release of odt2daisy (9Nov09)
- Works with OpenOffice.org 3.0 or later, on any platform
- Takes full advantage of multi-lingual encoding of Writer documents, and selects the appropriate text-to-speech engine for each language-tagged passage
- Takes full advantage of “styles”, document structure
- Flags badly formed documents to the user

- Next steps: assistance to document authors in fixing problems; perhaps even some “visualization”

ACCESSIBLE Impairment Simulator

- Pair of NetBeans plug-ins – to be available to all NetBeans developers
- Helps developers with four things needed to successfully & efficiently build accessible applications:
  - Empathy: visceral understanding of ***why*** any particular type of accessibility is needed
  - Knowledge of ***what*** is wrong (in their code); ***what*** their code should be done
  - Knowledge of ***where*** the problems are in their code
  - (Ideally) automated help fixing their code

National Broadband Plan thoughts

- Some key tasks:
  - Acquisition of accessible devices (e.g. PCs with AT)
  - Ongoing maintenance of accessible PCs
  - Training in how to use them
- Options for devices:
  - Windows with commercial AT
  - Macintosh with built-in & commercial AT (some built-in)
  - Open source AT (on Windows, or on UNIX)
  - Accessibility delivered “from the cloud”

Thoughts continued

- Shift from gov't paying twice, to more of "ÆGIS model"
- Invest in components that can be used multiple places, by multiple sorts of solutions
  - > E.g. OpenGazer
- Develop solutions that are supportable
  - > Story of support @ SSA for the ~1,500 disabled users...
- Collaborate with EC on accessibility research
- New and attractive idea: National Public Inclusive Infrastructure

Some things we like about the NPII

- Their statement of the problems
  - > AT cost & lack of AT/accessibility options
  - > Lack of awareness & training
  - > Importance of developer tools, authoring tools
  - > Need for R&D for under-served disabilities (e.g. cognitive)
- Focus on 3rd generation, on open source access
- Focus on building ecosystem of re-usable components
- Leveraging the cloud for access anywhere

Addressing Accessibility: 3rd generation, ÆGIS, and Sun's approach

Peter Korn
peter.korn@sun.com

What is FP7?

- FP7 is the Seventh EC Research Framework Programme, a massive, multi-year research effort
  - http://cordis.europa.eu/fp7/
  - Runs from 2007 – 2013
  - €50bn in funds total
  - €9.1bn for ICT
  - FP7 funds ***research*** in a variety of formats, vehicles
  - There were 6 previous “Framework Programmes”

What things does FP7 fund?

- Understanding FP7:
  http://ec.europa.eu/research/fp7/index_en.cfm?pg=understanding
- Funds research in 6 broad areas:
  > ***Cooperation***
  > Ideas
  > People
  > Capacities
  > Nuclear Research
  > Joint Research Centre
- We will focus on “Cooperation”, which is €32bn

What FP7 funds, in more detail

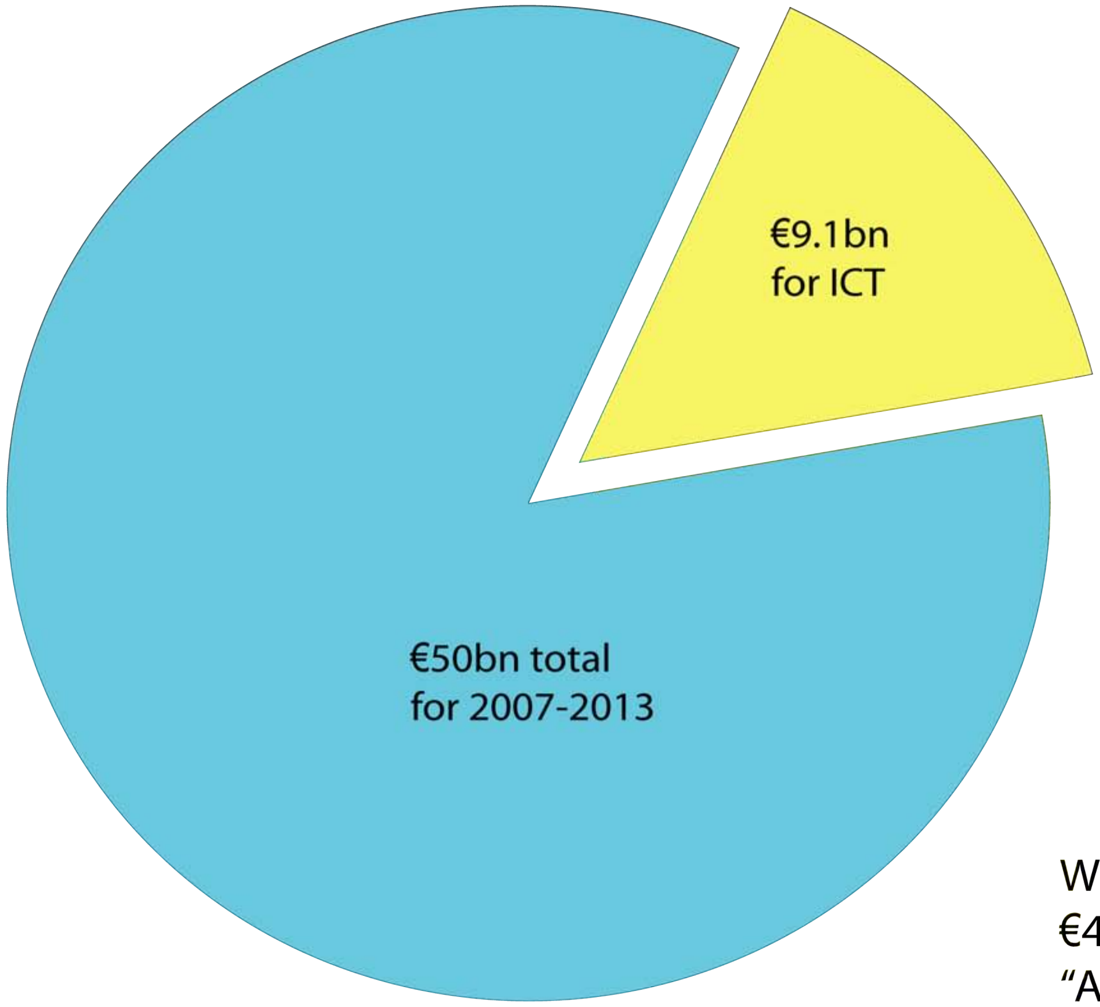
- "Cooperation" is broken down into 10 areas:
  - > Nano-production
  - > Environment
  - > Energy
  - > Transport
  - > Space
  - > Socio-economic sciences & humanities
  - > Security
  - > Health
  - > Food, Agriculture, biotech
  - > ***Information & Communication Technologies (ICT)***

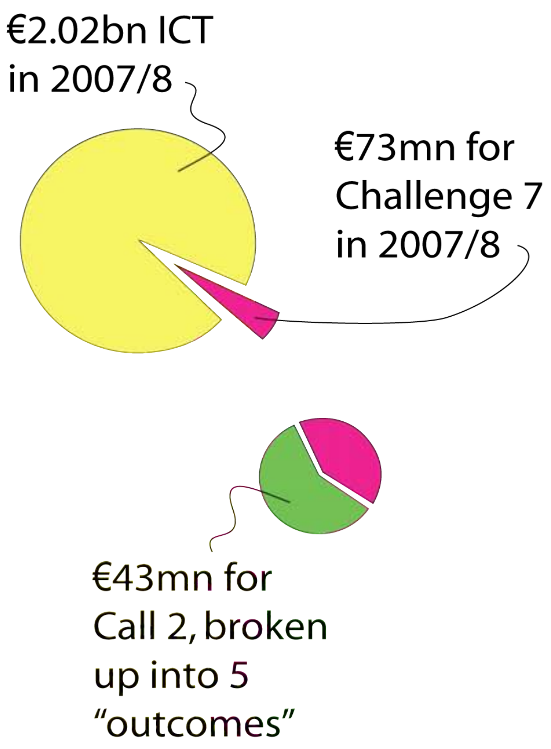
FP7 funding, in still more detail

- For computer technology, we are interested in ICT, which has €9.1bn funding 7 “challenges”:
  - Pervasive and Trustworthy Network and Service Infrastructures
  - Cognitive Systems, Interaction, Robotics
  - Components, systems, engineering
  - Digital Libraries and Conten
  - Towards sustainable and personalised healthcare
  - ICT for Mobility, Environmental Sustainability and Energy Efficiency
  - ICT for Independent Living, Inclusion and Governance

  See http://cordis.europa.eu/fp7/ict/ & the current work programme: ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/ict-wp-2009-10_en.pdf

FP7 – pieces of the pie...





We're looking at portion of the €43mn in ***ICT-2007.7.2*** for "Accessible & Inclusive ICT"

FP7 funding instruments

- Large scale collaborative (integrating) projects - “IP”
  - 10-20 “optimum” in consortia, €4-25m; 3-5 years
- Small or medium-scale focused research - “STREP”
  - 6-15 “optimum” in consortia, €1-4m, 1.5-3 years
- Network of Excellence - “NoE”
  - [[[need details here]]]
- Coordination Action - “CA”
  - 13-26 participants, €0.5-2m, 1.5-3 years
- Support Action - “SA”
  - 1-15 participants, €0.03-3m, 9-30 months

What & how FP7 pays for stuff

- Costs are paid as follows:
  - R&D costs: 50% match for large companies;
    75% match for others (Universities, small companies)
  - Demonstration activities: 50% match
  - “Other activities”: 100% (e.g. Consortium management)
  - Coordination & support actions: 100%
- Payment of large sums done as wire transfers
- Payment done on estimated costs, paid in advance
  - Period typically 12 mos. for cycle
  - At the end of each period, you show receipts, make estimate for next period, get payment in advance for that

Addressing Accessibility: 3rd generation, ÆGIS, and Sun's approach

Peter Korn
peter.korn@sun.com